1. What is command injectio

Command Injection is a type of security vulnerability that occurs when an attacker can execute arbitrary commands on a host operating system through a vulnerable application. This often happens when an application improperly processes user-supplied input, allowing the attacker to "inject" commands that the system then executes. Command injection is dangerous because it can lead to full system compromise, data theft, or other forms of abuse if not mitigated.

Key Points to Note:

1. **How it Works**:
   - Command injection typically occurs in applications that take user inputs and pass them to system commands directly (e.g., through shell commands) without properly validating or sanitizing the input.
   - For example, if an application takes a filename as input and runs `cat <filename>` directly, an attacker could potentially inject commands by using characters like `;` or `&&` to execute additional commands.
2. **Types of Command Injection**:
   - **Arbitrary Command Execution**: The attacker can execute any command they want on the system.
   - **Blind Command Injection**: The response isn't directly shown to the attacker, so they have to use timing, side effects, or other indirect methods to infer the outcome.
3. **Common Injection Points**:
   - Forms that accept text or filenames.
   - URL parameters that get passed to a command.
   - Cookies or headers that are parsed by scripts.
4. **Typical Injection Characters**:
   - Shell metacharacters like `;`, `|`, `&&`, and `||`.
   - In Windows, special symbols like `&`, `|`, or `^` could be used similarly.
5. **Example**:

```
# Intended command
ls /some/directory

# Injected command
ls /some/directory; rm -rf /
```

6. **Prevention Tips**:

  - Use secure coding practices: avoid directly passing user input into system commands.
  - Escape or sanitize inputs rigorously.
  - Implement **parameterized commands** or use **language-specific libraries** designed to handle OS interactions securely.